



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/598,713	06/21/2000	Jonathan P. Douglas	2207/8613	7081
23838	7590	11/23/2004	EXAMINER	
KENYON & KENYON 1500 K STREET, N.W., SUITE 700 WASHINGTON, DC 20005			HUISMAN, DAVID J	
			ART UNIT	PAPER NUMBER
			2183	
DATE MAILED: 11/23/2004				

Please find below and/or attached an Office communication concerning this application or proceeding.

<b>Office Action Summary</b>	Application No.	Applicant(s)
	09/598,713	DOUGLAS, JONATHAN P.
Examiner	Art Unit	
David J. Huisman	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --  
**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM  
 THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

#### **Status**

1) Responsive to communication(s) filed on 12 October 2004.  
 2a) This action is **FINAL**.                    2b) This action is non-final.  
 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

#### **Disposition of Claims**

4) Claim(s) 4-7,9-12,17-20,23,25 and 26 is/are pending in the application.  
 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.  
 5) Claim(s) \_\_\_\_\_ is/are allowed.  
 6) Claim(s) 4-7,9-12,17-20,23,25 and 26 is/are rejected.  
 7) Claim(s) \_\_\_\_\_ is/are objected to.  
 8) Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

#### **Application Papers**

9) The specification is objected to by the Examiner.  
 10) The drawing(s) filed on 15 August 2003 is/are: a) accepted or b) objected to by the Examiner.  
 Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
 Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).  
 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

#### **Priority under 35 U.S.C. § 119**

12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).  
 a) All    b) Some \* c) None of:  
 1. Certified copies of the priority documents have been received.  
 2. Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.  
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

#### **Attachment(s)**

1)  Notice of References Cited (PTO-892)  
 2)  Notice of Draftsperson's Patent Drawing Review (PTO-948)  
 3)  Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)  
 Paper No(s)/Mail Date \_\_\_\_\_

4)  Interview Summary (PTO-413)  
 Paper No(s)/Mail Date. \_\_\_\_\_

5)  Notice of Informal Patent Application (PTO-152)

6)  Other: \_\_\_\_\_

**DETAILED ACTION**

1. Claims 4-7, 9-12, 17-20, 23, and 25-26 have been examined.

***Papers Submitted***

2. It is hereby acknowledged that the following papers have been received and placed of record in the file: Amendment as received on 10/12/2004.

***Maintained Rejections***

3. Applicant has failed to overcome the prior art rejections set forth in the previous Office Action. Consequently, these rejections are respectfully maintained by the examiner and are copied below for applicant's convenience.

***Maintained Claim Rejections - 35 USC § 102***

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

5. Claim 10 is rejected under 35 U.S.C. 102(b) as being anticipated by Yoshioka et al., U.S. Patent No. 4,394,729 (herein referred to as Yoshioka).

6. Referring to claim 10, Yoshioka has taught a method for interfacing an instruction pipe with a return stack buffer having a predetermined round-trip communication latency period associated with a communication path therebetween, the method comprising:

- a) reading a return instruction from an instruction pipe stage. See Fig.11, and note a return instruction is read.
- b) determining, with reference to other instructions read previously from the instruction pipe stage, whether a return address is available to the instruction pipe prior to expiration of the round-trip communication latency period with the return stack buffer. See Fig.10 and Fig.11. When a return instruction is encountered, it will be determined whether a previous subroutine jump/call instruction has set BANKV (CNT) = 1 and whether CNT≤8. If BANKV (CNT) = 1 and CNT≤8, then the address is available prior to the expiration of the latency period associated with the return stack.
- c) if the return address is available immediately upon receipt of the return instruction at the instruction pipe stage, forwarding the return address to a next pipe stage during a next clock cycle, and if not, stalling processing of the return instruction until the round-trip communication latency period expires and forwarding a received return address thereafter. Again, see Fig.11 and column 9, lines 6-30. Note that if the address is immediately available (i.e., BANKV (CNT) = 1 and CNT≤8), the address will be read from the register and supplied for further processing in the instruction pipe. However, if the address is not in the register (BANKV (CNT) ≠ 1 or CNT>8), then an access must be made to the stack, which is in memory. This will require stalling since accessing memory is slow.

7. Referring to claim 12, Yoshioka has taught a method as described in claim 10. Furthermore, it is the inherent nature of a stall to stall the instruction pippetage and all other instruction pippetages before it in the instruction pipe. For instance, see Figure 3.13 on page 154 and note that if the SUB instruction is stalled before its 3<sup>rd</sup> stage (as shown), then the next two subsequent instructions are stalled before their 2<sup>nd</sup> and 1<sup>st</sup> stages respectively.

8. Claim 17 is rejected under 35 U.S.C. 102(e) as being anticipated by Pickett, U.S. Patent No. 5,968,169 (as applied in the previous Office Action).

9. Referring to claim 17, Pickett has taught execution logic for a processor, comprising:

- a) a first instruction pipe, comprising a first plurality of cascaded pippetages, the first instruction pipe having a return stack buffer. See Fig.15 and note the cascaded stages of the pipeline (a new stage per clock cycle). The first pipe would include one of the multiple decode units (decoder 208A for instance) and multiple functional units, shown in Fig.1, in order to operate on one many instructions fetched per clock cycle. See column 18, lines 31-38. Note that up to 6 instructions can be passed to 6 pipelines (Fig.1, components 208, 210, 212). Also, note that decoder 208A is coupled to the branch prediction unit 220, which according to Fig.2, comprises a return stack buffer 250. Therefore, the first pipe has and uses a return stack buffer.
- b) a second instruction pipe, comprising a second plurality of cascaded pippetages. Note that a second pipeline would include a second set among the decode units, functional units, etc, shown in Fig.1 (for instance, decoder 208B). This pipeline would also have pippetages as shown in Fig.15.

c) the second instruction pipe being in communication with the return stack of the first instruction pipe through a communication path having a communication latency that is different from the communication latency between the first instruction pipe and the return stack buffer.

Note that each pipeline decoder is coupled to the return stack buffer. See Fig.1, Fig.2, and column 11, lines 15-19, and note that each decoder (and pipeline) is in communication with the same return stack.. Furthermore, from Fig.1, it can be seen that the multiple decoders would all be located on different portions of the chip. More specifically, it is impossible for each decoder component to be located at the exact same physical location on the chip. Consequently, the length of the wires (communication paths) from each decoder to the return stack buffer will differ in length, and consequently, cause the communication latency to be different for the pipelines. Clearly, if the wires are a different length, then the latency will be of different length. On the other hand, even if the designers had tried to achieve the same length wires from each pipeline to the return stack buffer, there will still be some minor difference in the wire length (even if the difference is minute). Even the most minute difference in wire length will result in some latency difference. Therefore, based on the layout of the component on the chip and the imperfections in wire, the data in the wires will not travel the same distance in the same amount of time.

10. Claim 23 is rejected under 35 U.S.C. 102(b) as being anticipated by IBM Technical Disclosure Bulletin NN9204269 (as applied in the previous Office Action and herein referred to as IBM).

11. Referring to claim 23, IBM has taught an instruction control method, comprising, responsive to a return instruction in a first pipestage of an instruction pipe:

a) determining whether the pipestage processed a prior return instruction faster than a latency period for round trip communication between the pipestage and the return stack buffer, and if so, stalling the downstream pipestages until the period for processing a prior return instruction equals the round trip communication latency period. See page 2, and note the 5 steps. Also, see page 3, and note the description regarding a return instruction followed by another return instruction. In this situation, if the prior return instruction has not yet popped the stack (processed faster than a latency for communication with the RSB), then the subsequent return instruction must be stalled. Otherwise, the subsequent return instruction will retrieve an incorrect address from the stack, i.e., the address that should be popped off by the prior return instruction. As discussed on page 3 of IBM, the stalling will take place until the prior return instruction completes by popping the address off of the stack. At this time, the latency period for communicating with the stack will expire.

12. Claim 25 is rejected under 35 U.S.C. 102(b) as being anticipated by Armstrong, U.S. Patent No. 4,394,729.

13. Referring to claim 25, Armstrong has taught an instruction pipe, comprising:

a) a plurality of pipestages connected in cascade. See Fig.1, and note that fetch, decode (interpret), and execute stages are shown.

b) first and second registers provided between first and second pipestages of the plurality. See Fig.4A, for instance, and note the one register is the counter register and another register would

be the register at the top of the register file stack (return stack buffer). This would be a pair of registers.

- c) the first register to store a return address received from the pipestage during receipt of a call instruction. See Fig.4B-C and column 3, line 55, to column 4, line 33. Note that when a call instruction is encountered, a corresponding return address is pushed into the top-of-stack register.
- d) the second register to store a return address received from a return stack buffer. See Fig.4B-D and note the counter register. This counter register receives the next most recent return address from the return stack buffer. See column 4, lines 34-45.
- e) a selector coupling the first and second registers to the second pipestage. See Fig.1, component 36, for instance. This selector (multiplexer) is coupled to component 30, which as shown in Fig.2, comprises the return stack buffer 303 and the counter register. Therefore, the selector is coupled to both of these components.

***Maintained Claim Rejections - 35 USC § 103***

14. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

15. Claim 11 is rejected under 35 U.S.C. 103(a) as being unpatentable over Yoshioka, as applied above, in view of IBM, as applied above.

16. Referring to claim 11, Yoshioka has taught a method as described in claim 10. Yoshioka has not taught the specifics of claim 11. However, IBM has taught determining whether the

return instruction requires access to the return stack buffer in excess of an access allocation for the instruction pipe, and if so, stalling the return instruction. Looking at page 3 of IBM, it is explained that if two return instructions are close to each other, then the second will have to be stalled since only one should access the stack at a time. More specifically, the first return must be able to pop the stack before the second return reads from the stack. Otherwise, the second return will retrieve the wrong address. This is applicable to Yoshioka because if two a second return instruction is close to a first return instruction and it turns out that both need to retrieve addresses from the stack (external resource), then the second return address must be stalled until the first one pops the stack, thereby allowing the second return to read the correct address. As a result, to prevent corruption, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Yoshioka in view of IBM such that a return instruction is stalled if exceeds the access allocation for the instruction pipe.

17. Claims 18 and 19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Pickett, as applied above, in view of Sproch et al., U.S. Patent No. 6,247,134 (as applied in the previous Office Action and herein referred to as Sproch).

18. Referring to claim 18, Pickett has taught logic as described in claim 17. Pickett has not taught the specifics of claim 18. However, Sproch has further taught clock throttling logic which comprises:

a) a state machine coupled to an output of the at least one pipestage from the second plurality of pipestages. See Fig.5 and note that state machine 210 is coupled to the first stage of a pipeline. It determines whether to stall a pipeline or not (2-state state machine).

b) a clock control circuit having an input for a system clock signal and having an output for a modified clock signal, the output coupled to the at least one pipestage, the clock control circuit controlled by the state machine. See Fig.5 and see components 351, 352, 353, 355, 362, 363, and 365. These components represent the innards of component 230, shown in Fig.3. This circuitry takes in a clock signal and modifies the output clock signals based on the state machine circuit 210. Also, see column 8, lines 50-61.

This throttling circuitry takes in a clock signal and modifies the output clock signals based on the need for a stall. Also, see column 8, lines 50-61. The abstract of Sproch shows that such a concept allows for power saving within the pipeline. Therefore, in order to save power, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Pickett to include clock throttling logic as taught by Sproch. It should also be noted from Fig.5 that this logic is coupled to at least one pipestage of a pipeline.

19. Referring to claim 19, Pickett has taught logic as described in claim 17. Furthermore, claim 19 is the same as claim 18 except that the state machine is coupled to at least one of the first plurality of pipestages in claim 19 as opposed to the at least one of the second plurality of pipestages in claim 18. However, one of ordinary skill in the art would have recognized that if each pipeline were to include the ideas taught by Sproch, then each pipeline would be able to save power, resulting in more overall system power being saved. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Pickett to have a state machine coupled to the first plurality.

20. Referring to claim 20, Pickett has taught logic as described in claim 17. Pickett has further taught that additional instruction pipestages from either the first or the second instruction

pipe are provided in communication with the return stack buffer, the additional instruction pipe stages also provided with additional clock throttling logic. Recall from above that the decode units (which operate in decode stages of pipelines) communicate with the RSB via selector 258 as shown in Fig.2. In addition, as shown in Fig.2 and Fig.4, when a return address is selected from the stack it is applied to the instruction cache as a fetch address. Therefore, the RSB is also in communication with the fetch stage of the pipelines since a return address is a fetch address. In addition, it should be noted from Sproch that the clock throttling logic, which is used to save power, is coupled to each of the pipeline stages. Consequently, in order to save power, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Pickett to include clock throttling logic as taught by Sproch.

21. Claim 26 is rejected under 35 U.S.C. 103(a) as being unpatentable over Armstrong, as applied above, in view Sproch, as applied above.

22. Referring to claim 26, Armstrong has taught an instruction pipe as described in claim 25. Armstrong has not explicitly taught a clock stopping circuit to control the second pipe stage and pipe stages downstream therefrom. However, Official Notice is taken that instruction dependencies and the stalls that result therefrom, are well known and accepted in the art. It is well known that throughout an executing program, instructions are dependent on prior instructions. If a given prior instruction has not finished in time to satisfy the dependency of a subsequent dependent instruction, then the dependent instruction must be stalled until the dependency can be satisfied. And, it is well known that to stall an instruction from progressing through the pipeline, the appropriate pipeline stages should not be pulsed with a clock signal.

This scheme is shown in Sproch. See Fig.3, for instance, and note that each pipeline stage receives a clock signal, whereby when the clock is received the data from each stage is passed to the next. However, if a stall condition is encountered, a clock signal will not be applied to the appropriate stage, thereby preventing data from that stage from moving to the next stage. Furthermore, it is well known in the art of stalling that when one stage is stalled, all stages prior to that stage are also stalled (downstream stages). Without a scheme similar to that of Sproch to effect stalls within the pipeline, data can be corrupted due to dependent instruction executing when their dependencies have not been satisfied. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to control the second pipestage and pipestages downstream via clock stopping circuit, as shown in Sproch.

#### *Response to Arguments*

23. Applicant's arguments filed on October 12, 2004, have been fully considered but they are not persuasive.
24. Applicant argues the novelty/rejection of claim 10 on page 7 of the remarks, in substance that:

“...the condition  $\text{BANKV}(\text{CNT})=1$  and  $\text{CNT} \leq 8$  has nothing to do with the round trip communication latency period. In fact, Yoshioka says nothing about the timing problem that arises due to round trip communication latencies between the instruction pipe and the RSB and/or maintaining timing synchronism between the instruction pipe and the RSB”

“Yoshioka says nothing about forwarding the return address to a next pipestage and stalling processing of the return instruction.”

25. These arguments are not found persuasive for the following reasons:
  - a) Regarding the first argument, the examiner asserts that the aforementioned condition has everything to do with the round trip communication latency period. Looking at Fig.11, and

column 9, lines 6-30, it can be seen that if the condition is true, then the return address is stored within a pipeline register, whereas if the condition is false, the return address is stored in the memory return stack (RSB). More specifically, if the address is in the RSB, there will be a latency associated with retrieving that address, as an access to main memory is costly. On the other hand, if the address is in the register, then the address is available prior to the time when it would've been retrieved from the RSB. That is, if the retrieval of the return address begins at time X, then if the address is in the register, it will be available at time X+N. However, if the address is in the RSB, it will be available at time X+M, where M>N.

b) Regarding the second argument, it is clear that the return address must be forwarded to a next pipestage. It is inherent that in a pipeline, an instruction and its associated data progress through a pipeline, stage by stage. If they did not move through the pipeline, then the instruction would never execute. Consequently, according to claim 11, when the instruction is decoded and the return address is read, it will have to be passed on to a next pipestage, i.e., the execute stage, where the return address will be inherently written to the program counter (the return instruction cannot be executed without the return address...therefore, once the return address is retrieved, it must be passed to the execute stage so that the return instruction can execute). This also shows that processing of the return instruction will be stalled until the latency expires. As just previously mentioned, the return instruction cannot execute until the return address is available. Consequently, if the return address must be retrieved from the memory stack, then the processing of the return instruction will be stalled until the time to retrieve the address from the memory stack expires. This is why Yoshioka implements the pipeline register. By having it in a register, a memory stack would not have to be accessed, thereby reducing the stall time.

26. Applicant argues the novelty/rejection of claim 17 on page 8 of the remarks, in substance that:

“Pickett does not disclose a return stack buffer being included in the first instruction pipe and also in communication with a second instruction pipe. Rather, Pickett’s return stack buffer is included in the branch prediction unit 220. Neither the multiple decode units nor the multiple functional units, which are alleged to be equivalents of the first and second instruction pipes by the Office Action.”

27. These arguments are not found persuasive for the following reasons:

a) Fig. 1 and Fig. 2 of Pickett show that the first pipe (i.e., with decoder 208A, for instance) has access to a return stack buffer 250 (RSB). This RSB (in the branch prediction unit) can be considered part of the pipeline as the pipeline needs to access it for branch/return type instructions. Also, it should be realized that applicant has merely claimed that the pipeline has an RSB. This is very broad, as I can say “I have lungs” (meaning the lungs are within me) or I can say “I have an apple” (meaning I have access to an external object). The examiner asserts that either way, Pickett anticipates the current claim language.

28. Applicant argues the novelty/rejection of claim 23 on page 9 of the remarks, in substance that:

“IBM does not disclose, teach, or suggest this subject matter of claim 23. IBM states that “if a return is followed by another return instruction before the first one completes, you need to hold the second return in decode until the first one completes in write back stage.” This “before the first one completes” refers to execution time of a return instruction (or the time it takes to execute a return instruction), but not the round-trip communication latency period with the return stack buffer. Thus, IBM fails to disclose, teach, or suggest stalling the pippetages until the period for processing a prior return instruction equals the round trip communication latency period.”

29. These arguments are not found persuasive for the following reasons:

a) The examiner asserts that accessing the return stack and suffering the round-trip communication latency is part of return instruction execution. That is, when a return instruction is to be executed, the stack must be popped, i.e., the stack must be accessed, thereby forcing the system to spend time accessing the stack. The time spent is the round-trip communication latency (the total time between sending a “retrieve” signal to the stack and actually obtaining the return address). From step 3 of 5 in the IBM document (page 2), it is disclosed that the popping of the stack (retrieval of the return address) is done during the write-back stage while pus. The, on page 3, if a return follows a return, the second one must be stalled until the first one is done retrieving, otherwise corruption will occur.

30. Applicant argues the novelty/rejection of claim 25 on page 10 of the remarks, in substance that:

“As illustrated in Fig.1, the JRS is not provided between the first and second pipestages of the plurality. Thus, Armstrong fails to disclose first and second registers provided between first and second pipestages of the plurality.”

“Additionally, Armstrong does not disclose, teach, or suggest “the first register to store a return address received from the first pipestage during receipt of a call instruction and the second register to store a return address received from a return stack buffer.” Rather, Armstrong merely discloses that the counter/register always stores the latest entry into the top of the memory stack so that it is immediately available to the control register. Thus, Armstrong fails to disclose, teach, or suggest the subject matter of claim 25.

31. These arguments are not found persuasive for the following reasons:

a) Regarding the first argument, it is not clear how Fig.1 of Armstrong does not show that the first and second registers are in between pipestages. Firstly, Fig.1 is simply a drawing and JRS could have been drawn in some different location. Secondly, applicant should realize that the physical location of the JRS does not have to be between pipestages. Instead, it may function as if it were between pipestages. For instance, the JRS ultimately receives the return address from

the CRF in the fetch stage (first pipestage). And, the return address is ultimately sent to the execute stage (second pipestage). See Fig.1. Consequently, the JRS is between pipestages.

b) Regarding the second argument, the examiner asserts that this argument supports the examiner's position. That is, when a call is executed, a return address originating in the first pipestage (fetch stage) is ultimately pushed into the top register in the register file stack. See Fig.4B and Fig.4C and column 3, line 55, to column 4, line 33. Then, when return instructions are executed, the top register is copied into the counter register (second register) so that when the next return instruction executes, the corresponding return address will be quickly retrieved.

### *Conclusion*

32. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

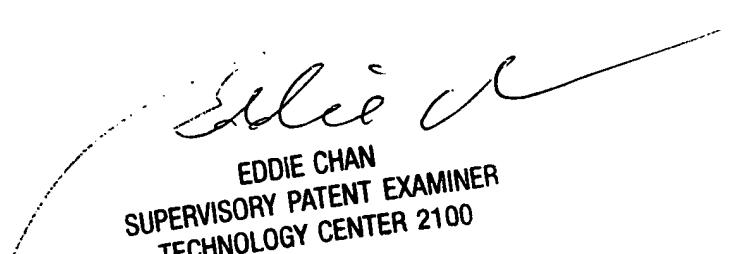
A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to David J. Huisman whose telephone number is (571) 272-4168. The examiner can normally be reached on Monday-Friday (8:00-4:30).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

DJH  
David J. Huisman  
November 18, 2004

  
EDDIE CHAN  
SUPERVISORY PATENT EXAMINER  
TECHNOLOGY CENTER 2100